**REGULAR ARTICLE**

# Optimizing Flow Shop Scheduling for Energy Efficiency and Sustainability in Semiconductor Manufacturing: A Comparative Study of NEH and PIG_NEH Algorithms

Nameet Kumar Sethy[1,*], Dhiren Kumar Behera[2]

[1] *Department of MECH, Indira Gandhi Institute of Technology, Sarang, BPUT, Rourkela, Odisha, India*
[2] *Department of PROD, Indira Gandhi Institute of Technology, Sarang, BPUT, Rourkela, Odisha, India*

(Received 12 August 2025; revised manuscript received 16 October 2025; published online 30 October 2025)

This work contributes to an important challenge faced by the semiconductor backend manufacturing industry with high operational complexity and energy consumption. Traditional NEH scheduling algorithms minimize makespan but do not consider the energy and sustainability objective. To address this, we introduced and evaluated a new hybrid algorithm, PIG_NEH (Population-based Iterated Greedy NEH), which incorporates iterative refinement and population based search methodologies. The method is based on computer simulations implemented in Python (v3. 11) on a Windows 11 machine powered by a 2.41GHz AMD Ryzen 3 CPU paired with 4GB of RAM. The experiments, through the use of static data, saw evaluations of scheduling performance for five distinct parameter configurations, including four (4) datasets with sizes of 8, 20, 30 and 50, using 5 and 20 machines. We evaluated metrics like makespan, energy efficiency and computational time. Statistical comparisons were made between NEH and PIG_NEH and shown as t-tests to highlight trade-offs. PIG_NEH reduces makespan by a further 1.85% over NEH as well as increases energy efficiency although a 37,622.54 % computational overhead is attained. These results contribute to sustainable scheduling practices in two ways, with the first being improved utilization of resources and the second being the integration of energy-saving goals into the manufacturing systems.

**Keywords:** NEH, PIG_NEH, Make span, Flow-shop, Scheduling, Back-end semiconductors manufacturing.

DOI: 10.21272/jnep.17(5).05032 PACS numbers: 07.05.Tp, 73.61.Jc

## 1. INTRODUCTION

Scheduling evolved from handling simple tasks to solving high-end algorithms for the industrial evolution. It is the intended allocation of resources to achieve quality, cost, and sustainability targets. For example, in backend semiconductor manufacturing, flow shop scheduling has been optimized to make wafer testing, assembly, and packaging activities more efficient to minimize time and lead time losses, resulting in greater productivity. This also optimizes the use of resources, reduces operational costs, and avoids environmental degradation. Modern scheduling balancing efficiency with sustainability aims to reduce makespan and energy consumption to meet worldwide sustainability targets. Existing algorithms including NEH have limitations for being energy-efficient or scalable in computation for these energy-intensive industries that play a crucial role in sustainable manufacturing. In this paper, we introduce a method to include energy efficiency in scheduling scenarios. The same applies to renewable energy, where advanced scheduling is critical for the scheduling of power generation, grid operations, and system maintenance. Only few studies examine the trade between computational efficiency and energy conservation of complex systems.

Designing an optimal schedule to execute a set of jobs on m machines has been a classic and a very important problem in operations research and scheduling theory, also known as the static permutation flow shop sequencing problem (PFSP) to minimize makespan, where we have to process a set of n jobs to m machines. First proposed in 1983, the Nawaz–Enscore–Ham (NEH) algorithm gives a simple heuristic approach that generates job sequences based on total processing time accumulated by scheduling jobs. Recent advancements in the Nawaz-Enscore-Ham (NEH) algorithm and its derivatives have significantly expanded its applicability to complex scheduling problems. Chen and Wu (2022) [1] improved NEH to handle setup times, enhancing its utility for constrained scenarios. Li and Zhang (2024) [2] integrated NEH with discrete-event simulation for high-precision satellite scheduling. Paredes-Astudillo et al. (2023) [3] combined NEH with local search operators for learning-effect scheduling. Ribas and Mateo (2010) [4] enhanced NEH with reversibility properties and tie-breaking strategies. Zhao et al. (2022) [5] integrated NEH with Q-learning and artificial bee colony methods for better task initialization, while Singh et al. (2022) [6] introduced the VN-NEH+ algorithm to improve efficiency. Amri Abu Bakar (2022) [7] applied machine learning to automate NEH in semiconductor manufacturing. Liu, Wang, and Jin (2007) [8] developed a PSO-based memetic algorithm for complex scheduling. Wang et al. (2023) [9] proposed N-NEH+ with swap methods to enhance scalability and accuracy. Puka et al. (2022) [10] studied input sequence effects on NEH performance.

* Correspondence e-mail: nameet158@gmail.com





Cite this article as: N.K. Sethy et al., *J. Nano- Electron. Phys.* 17 No 5, 05032 (2025) https://doi.org/10.21272/jnep.17(5).05032

Ruiz, Serifoglu, and Urlings (2008) [11] modeled realistic hybrid flexible flowshop scheduling problems, emphasizing practical applications and complexity. These developments showcase NEH's evolution and versatility in addressing modern scheduling challenges across industries.

This manuscript proposes the Population-based Iterated Greedy NEH (PIG_NEH) algorithm, a novel hybrid approach aimed at minimizing makespan and energy consumption through iterative refinement and population-based search strategies. The novelty lies in integrating energy-saving objectives into the scheduling process, reducing idle times and resource wastage, while addressing computational complexity and scalability challenges. This approach advances sustainable manufacturing by balancing computational and operational performance. The study significantly contributes to integrating scheduling algorithms into electricity-intensive semiconductor manufacturing, addressing energy conservation in flow shop scheduling. The findings also promote sustainability in operational processes beyond the studied system, supporting industrial optimization without compromising performance.

## 2. PROBLEM STATEMENT &METHODOLOGY

Existing scheduling algorithms, including NEH and its derivatives, focus on minimizing makespan but overlook energy conservation and computational complexity. This gap hinders the development of sustainable, efficient flow shop scheduling systems for energy-intensive industries like semiconductor manufacturing.

### 2.1. Research Objectives

This study aims to:
For the flow shop scheduling problem, compare the performance of NEH and PIG_NEH for minimizing makespan. Balancing execution costs with potential energy savings Introductions of a hybrid location based PIG_NEH algorithm which solves the efficiency and scalability problem. Build momentum for sustainable scheduling practices by embedding energy conservation in the workflows of manufacturing.
Given:
$n$: Number of jobs.
$m$: Number of machines.
pij: Processing time of job $j$on machine$i$
Objective:
Makespan:

$$\text{Minimize Cmax} = \text{maxjCmj}$$

where Cmj is the completion time of the last job j on the last machine m.

Assumptions:

1. Each job has a fixed processing time for each machine, given in the processing times matrix.
2. Each job incurs a waiting time between consecutive machines, specified in the waiting times matrix.
3. Each machine can only process one job at a time.
4. Jobs are processed sequentially on the machines.
5. A job that has commenced on machine i must be finished prior to the initiation of another job on that same machine.
6. There is no limit on the buffer capacity between machines.
7. If job j comes before job k on machine 1, it must also come before job k on all other machines.

### 2.2. NEH Algorithm

One popular heuristic method for reducing makespan in work scheduling issues, especially in flow shop scheduling settings, is the NEH (Nawaz, Enscore, and Ham) algorithm. It is well known for being straightforward and efficient in resolving combinatorial optimization issues.

Calculate Total Processing Time for Each Job: For each job $j$, compute its total processing time across all machines:

$$Tj = \sum_{i=1}^{m} pij \;\; \forall j \in \{1,2,\ldots,n\} \tag{1}$$

where $Tj$ represents the total workload of job $j$.

Sort Jobs by Total Processing Time: Sort the jobs in descending order based on Tj, creating an ordered list:

$$\text{Order Jobs: } T(1) \geq T(2) \geq \cdots \geq T(n) \tag{2}$$

Let this ordered list of jobs be denoted as $J = [j1, j2, \ldots, jn]$.

Initialize the Schedule: Start with the first job in the sorted list:

$$S = [j1] \tag{3}$$

Iterative Job Insertion: For each subsequent job $jk$ ($k = 2, 3,\ldots, n$), perform the following:
Test all possible positions for $jk$ in the current sequence $S$.
For each position $l(l = 1, 2,\ldots, |S| + 1)$:
Create a temporary sequence

$$S' = [S1,\ldots, Sl-1, jk, Sl,\ldots, S|S|] \tag{4}$$

Calculate the makespan $C\max(S')$ for this sequence:

$$C\max(S') = \frac{max}{i,j}\{Cij\} \tag{5}$$

where $Cij$ is the completion time of job $j$ on machine $iii$, calculated as:

$$\text{Cij}\begin{cases} \text{Ci} - 1,\text{j} + \text{pij} \\ Ci,j-1+pij,, \\ max(Ci-1,j,Ci,j-1)+pij, \end{cases} \tag{6}$$

Select the position $l$* that minimizes $C\max(S')$.

Update the schedule s .Repeat Until All Jobs Are Scheduled: Continue the insertion process until all n jobs have been added to the schedules.

### 2.3. PIG_NEH Algorithm

PIG_NEH An Enhanced Iterated Greedy Algorithm for PFSPs, improves NEH by leveraging population-based search and iterative refinement to minimize makespan efficiently. PIG_NEH enhances the NEH heuristic with iterated greedy methods to address sequence-dependent setup times, dynamic priorities, and scalability. It excels in large scheduling domains, including manufacturing, semiconductor processing, and logistics, improving performance and solution quality.

## 3. IMPLEMENTATION

Python (v3. 11) on a Windows 11 laptop powered by 2.41GHz AMD Ryzen 3 CPU and 4GB RAM. Libraries such as NumPy allowed for effective matrix manipulation, flowshop scheduling and numerical calculations. Problem instances in backend semiconductor manufacturing were analyzed using static data (e.g., job processing and machine idle times). Energy efficiency was

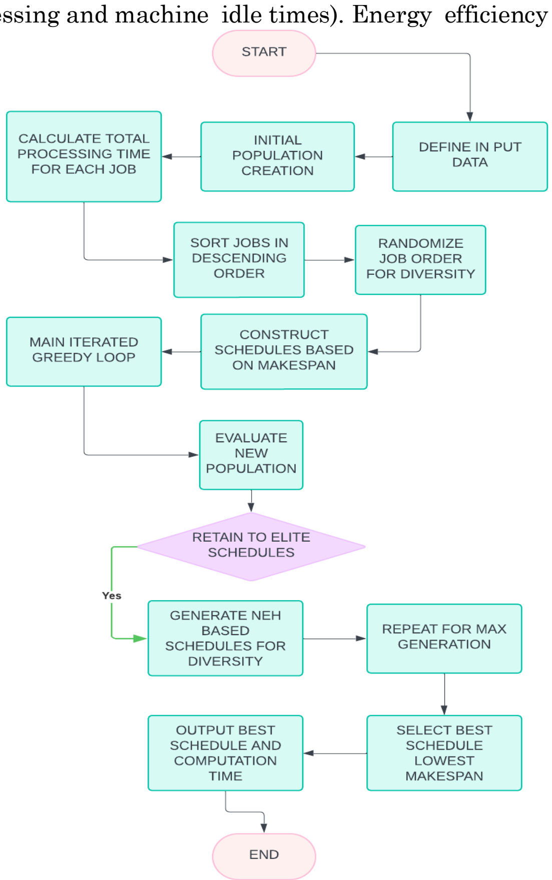


**Fig. 1** – Flowchart of Processes of PIG_NEH algorithm

also increased by optimized schedules for some processes such as wafer dicing, testing and packaging, which reduced idle times. NEH tested 10 iterations on 4 datasets and 20 machines to measure makespan and energy savings. Our experiments confirmed the Eulerian reduction of PIG_NEH evolution of the makespan and resources used by PIG_NEH compared with NEH.

## 4. COMPUTATIONAL EXPERIMENTS

Makespan denotes the overall duration required to finish a sequence of tasks on a machine, where reduced makespan values signify enhanced performance. Two heuristic scheduling approaches, NEH, PIGNEH are compared based on makespan, computation time and best schedule for different machine and task counts. The experiment aimed to minimize makespan using NEH, PIG_NEH algorithms across 8, 20,30 ,50 jobs on 5, and 20 machines.

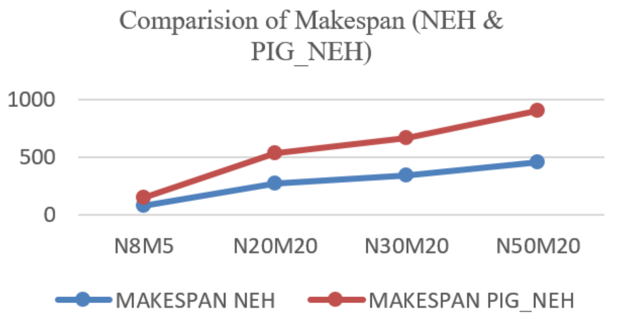


**Fig. 2** – Comparison of Makespan between NEH & PIG_NEH

PIG_NEH outperforms NEH with lower makespan, especially for larger problems, showcasing its scalability and real-world efficiency gains.

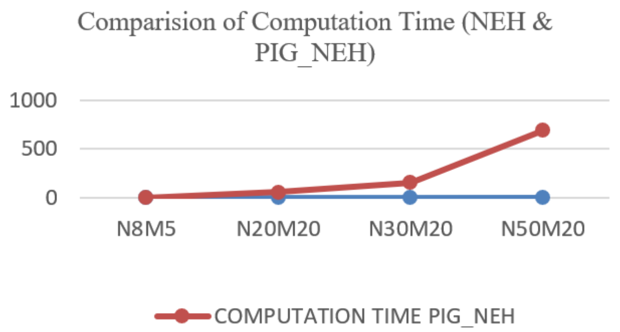


**Fig. 3** – Comparison of Computation Time between NEH & PIG_NEH

PIG_NEH is significantly slower than NEH, with computation time increasing exponentially as problem size grows, unlike NEH's faster, simpler heuristic. While PIG_NEH achieves better makespan results, its high computational cost limits its practicality for large-scale, time-sensitive problems.

## 5. STATISTICAL ANALYSIS & DISCUSSION

The makespan comparison is too close to show a significant difference as average percentage error is 1.85 % with standard deviations of 158.62 for NEH and 156.06 for PIG_NEH, but without significant difference ($p$ = 0.156). In contrast to the M/M/1 models, where computing times averages and standard deviation were close to each other, our exponents models show large differences in computation times, with the average %

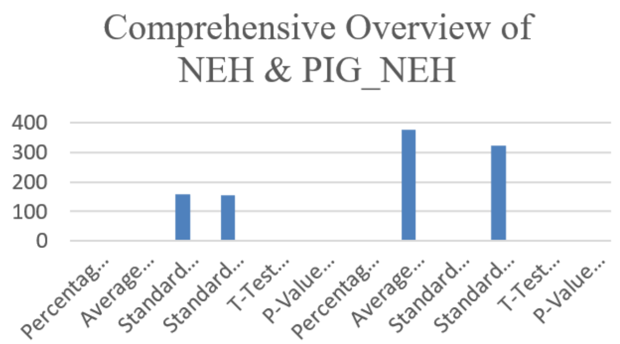


**Fig. 4** – Comparative Analysis of Makespan and Computation Time for NEH and PIG_NEH Algorithms

error of 37,622.54 % and other significant deviations (0.7225 – NEH, 324.39 – PIG_NEH). This confirms that the *t*-test p = 0.021 shows that the computational cost of PIG_NEH is significantly higher. These results show a trade-off between computational effort and better scheduling performance.

## 6. CONCLUSION

This research assessed the NEH and PIG_NEH algorithms for permutation flow shop scheduling in backend semiconductor manufacturing. PIG_NEH attained a reduced makespan, exhibiting an average percentage error of 1.85%; nevertheless, the difference was not statistically significant (p-value = 0.156). Nonetheless, PIG_NEH necessitated considerably greater computational time, with an average percentage error of 37,622.54% and a statistically significant p-value of 0.021. Notwithstanding its elevated computing expense, PIG_NEH exhibited enhanced energy efficiency by reducing machine idle periods and optimizing resource allocation, rendering it appropriate for sustainable manufacturing practices.

### 6.1. Discoveries

PIG_NEH surpassed NEH in reducing makespan, demonstrating moderate enhancements (average percentage error: 1.85%) and superior scalability for intricate situations. The computation times for PIG_NEH were markedly elevated, exhibiting an average percentage error of 37,622.54%, as corroborated by a statistically significant t-test (p-value: 0.021).

### 6.2. Recommendation

The PIG_NEH algorithm has broad applicability across various industries, including automotive manufacturing, logistics, healthcare, and energy utilities. It can optimize assembly lines by reducing downtime and managing sequence-dependent setup times, while in logistics, it can schedule delivery routes and warehouse operations efficiently, even accommodating real-time changes. In healthcare, PIG_NEH can balance energy usage and minimize wait times for surgeries or diagnostic procedures. Energy utilities can use it to optimize maintenance schedules for renewable energy systems and power grid operations. The algorithm is also adaptable for diverse challenges, such as dynamic scheduling in e-commerce or manufacturing, multi-objective optimization considering cost and emissions, and high-complexity systems like aerospace or telecommunications. Enhancements such as integration with machine learning for predictive modelling, hybridization with other heuristics, and energy-centric customization further extend its utility for energy-intensive and scalable applications.

## 7. NOVELTY

This paper introduces the application of NEH and the novel Population-based Iterated Greedy NEH (PIG_NEH) algorithm to address scheduling challenges in backend semiconductor production. PIG_NEH incorporates energy-saving objectives, reducing idle time and resource wastage to enhance sustainability. Statistical analysis using t-tests and p-values highlights trade-offs between computational time and performance. The novelty lies in PIG_NEH's iterative refinement and scalability, making it suitable for complex tasks, while NEH remains effective in computationally constrained environments. This study advances industry-specific scheduling by integrating energy efficiency and sustainability into optimization frameworks.

## REFERENCES

1. X. Chen, J. Wu, *Algorithms* **13** No 5, 112 (2022).
2. Y. Li, H. Zhang, *Appl. Sci.* **14** No 21, 9755 (2024).
3. E. Paredes-Astudillo, et al., *Springer Series on Scheduling Problems* (2023).
4. I. Ribas, M. Mateo, *Lecture Notes in Computer Science* **6073**, 44 (2010).
5. F. Zhao, et al., *IEEE Transactions on Industrial Informatics* (2022).
6. R. Singh, et al., *SSRN Electronic Journal* (2022).
7. Amri Abu Bakar, *IEEE 39th International Electronics Manufacturing Technology Conference (IEMT)*, 1 (2022).
8. B. Liu, L. Wang, Y.H. Jin, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **37** No 1, 18 (2007).
9. L. Wang, et al., *IEEE Transactions on Systems, Man, and Cybernetics* **54** No 3, 1633 (2023).
10. R. Puka, J. Duda, Stawowy, *Management and Production Engineering Review* **13** No 1, 32 (2022).
11. R. Ruiz, F.S. Serifoglu, T. Urlings, *Computers & Operations Research* **35** No 4, 1151 (2008).

**Оптимізація планування поточного цеху для енергоефективності та сталого розвитку у виробництві напівпровідників: порівняльне дослідження алгоритмів NEH та PIG_NEH**

Nameet Kumar Sethy[1], Dhiren Kumar Behera[2]

[1] *Department of MECH, Indira Gandhi Institute of Technology, Sarang, BPUT, Rourkela, Odisha, India*
[2] *Department of PROD, Indira Gandhi Institute of Technology, Sarang, BPUT, Rourkela, Odisha, India*

Ця робота робить внесок у вирішення важливої проблеми, з якою стикається галузь виробництва напівпровідникових серверних частин з високою операційною складністю та енергоспоживанням. Традиційні алгоритми планування NEH мінімізують тривалість виконання, але не враховують цілі енергоспоживання та сталого розвитку. Щоб вирішити цю проблему, ми представили та оцінили новий гібридний алгоритм PIG_NEH (Population-based Iterated Greedy NEH), який включає методології ітеративного уточнення та пошуку на основі популяції. Метод базується на комп'ютерному моделюванні, реалізованому на Python (версія 3.11) на машині з Windows 11, що працює на процесорі AMD Ryzen 3 з тактовою частотою 2,41 ГГц у парі з 4 ГБ оперативної пам'яті. Експерименти, що використовували статичні дані, включали оцінку продуктивності планування для п'яти різних конфігурацій параметрів, включаючи чотири (4) набори даних розміром 8, 20, 30 та 50, з використанням машин з 5 та 20. Ми оцінили такі показники, як тривалість виконання, енергоефективність та час обчислення. Було проведено статистичні порівняння між NEH та PIG_NEH, які представлені у вигляді t-тестів для виявлення компромісів. PIG_NEH зменшує тривалість виконання ще на 1,85% порівняно з NEH, а також підвищує енергоефективність, хоча досягається зменшення обчислювальних витрат на 37 622,54 %. Ці результати сприяють практиці сталого планування двома способами: перший – покращене використання ресурсів, а другий – інтеграція цілей енергозбереження у виробничі системи.

**Ключові слова:** NEH, PIG_NEH, Флоу-цех, Планування, Виробництво напівпровідників на серверній частині.